

99-291

Y&R  
JC892 U.S. P.O.  
09/75/065  
01/08/01

日本国特許庁  
PATENT OFFICE  
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日  
Date of Application:

2000年 1月 6日

出願番号  
Application Number:

特願2000-001043

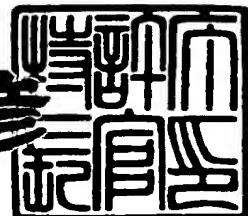
出願人  
Applicant(s):

インターナショナル・ビジネス・マシーンズ・コーポレイション

2000年 2月 25日

特許庁長官  
Commissioner,  
Patent Office

近藤 隆彦



出証番号 出証特2000-3011503

【書類名】 特許願

【整理番号】 JP9990291

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 12/00

【発明者】

【住所又は居所】 神奈川県大和市下鶴間1623番地14 日本アイ・ビ  
ー・エム株式会社 大和事業所内

【氏名】 片岡 克尚

【特許出願人】

【識別番号】 390009531

【住所又は居所】 アメリカ合衆国10504、ニューヨーク州アーモンク  
(番地なし)

【氏名又は名称】 インターナショナル・ビジネス・マシーンズ・コーポレ  
イション

【代理人】

【識別番号】 100086243

【弁理士】

【氏名又は名称】 坂口 博

【選任した代理人】

【識別番号】 100091568

【弁理士】

【氏名又は名称】 市位 嘉宏

【手数料の表示】

【予納台帳番号】 024154

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

特2000-001043

【包括委任状番号】 9706050

【包括委任状番号】 9704733

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 アプリケーションの起動方法、アプリケーションの起動のためのソフトウェア・プロダクト

【特許請求の範囲】

【請求項1】

サーバからロードしたアプリケーションをクライアント・マシン上で起動する方法であって、

(a) 前記クライアント・マシン側でのアプリケーション起動要求に応答して、サーバより実行環境を確認する実行環境確認アプレットをロードする段階と、

(b) 前記実行環境確認アプレットを実行した結果に基づいて、前記クライアント・マシン上で起動するアプリケーションが必要とするコードと前記アプリケーションの起動コマンドをサーバからロードする段階と、

(c) 前記起動コマンドを前記クライアント・マシンにおいて実行することにより、前記アプリケーションを起動する段階と、

を含むアプリケーションの起動方法。

【請求項2】

サーバからロードしたアプリケーションをクライアント・マシン上で起動する方法であって、

(a) 前記クライアント・マシン側でのアプリケーション起動要求に応答して、前記クライアント・マシン上で起動するアプリケーションが必要とするコードと前記アプリケーションの起動コマンドをサーバからロードする段階と、

(b) 前記起動コマンドを前記クライアント・マシンにおいて実行することにより、前記アプリケーションを起動する段階と、

を含むアプリケーションの起動方法。

【請求項3】

前記必要とするコードが前記クライアント・マシンに存在するか否かを判断する段階をさらに含み、前記アプリケーションが必要とするコードと前記アプリケーションの起動コマンドは、前記必要とするコードが前記クライアント・マシンに存在していない場合に前記必要とするコードをサーバからロードすることを特

徴とする請求項1または2に記載のアプリケーションの起動方法。

【請求項4】

サーバからロードしたアプリケーションをクライアント・マシン上で起動するためのソフトウェア・プロダクトであって、

(a) 前記クライアント・マシン側でのアプリケーション起動要求に応答して、サーバより実行環境を確認するための実行環境確認ソフトウェアと、

(b) 前記クライアント・マシン上で起動するアプリケーションが必要とするコードを取得する必要コード取得ソフトウェアと、

(c) 前記アプリケーションの起動コマンドを前記クライアント・マシンにおいて実行することにより、前記アプリケーションを起動するアプリケーション起動ソフトウェアと、

を含むソフトウェア・プロダクト。

【請求項5】

サーバからロードしたアプリケーションをクライアント・マシン上で起動するためのソフトウェア・プロダクトであって、

(a) 前記クライアント・マシン側でのアプリケーション起動要求に応答して、前記クライアント・マシン上で起動するアプリケーションが必要とするコードと前記アプリケーションの起動コマンドをサーバからロードする必要コード取得ソフトウェアと、

(b) 前記起動コマンドを前記クライアント・マシンにおいて実行することにより、前記アプリケーションを起動するアプリケーション起動ソフトウェアと、を含むソフトウェア・プロダクト。

【請求項6】

前記必要コード取得ソフトウェアは、前記必要となるコードが前記クライアント・マシンに存在するか否かを判断する機能を有していることを特徴とする請求項4または5に記載のソフトウェア・プロダクト。

【発明の詳細な説明】

【0001】

【産業上の利用分野】

この発明は、アプリケーションの起動方法に関するものであり、より詳しくは、サーバからロードしたアプリケーションをクライアント・マシン上で起動する方法に関するものである。

#### 【0002】

##### 【従来の技術】

従来より、ウェブ・ページ上に、直接インストーラの実行ファイル（exeファイル）へのリンクを置き、ユーザーに直接実行させる事でローカルへのアプリケーションの導入を行なっていた。しかし、この方法では、プラットフォームや条件により導入すべきパッケージが異なる場合など、どの導入パッケージを実行すべきかの判断や、前提ソフトウェアの確認をユーザー自身が行わなくてはならない為、最初の起動が繁雑であった。

#### 【0003】

この一方、アプレットは、配布の容易であるが、アプレットはブラウザの中で動作するという制限がある。一般に「署名付きアプレット」と呼ばれる第三者である認証機関が認証するアプレットも存在するが、署名付きアプレットに対するセキュリティの制限を解除したとしても、ウィンドウ・サイズの変更への追従や、終了を自分で制御できない等の制約がある。また、複数の署名付きアプレットを使用する場合、そのデジタル署名の管理も難しくなるという問題が発生する。

#### 【0004】

なお、公知技術として、特開平9-167085号公報が存在する。特開平9-167085号は、利用者が必要となるプログラム群をネットワークから自動的に取得するに際し、サーバ計算機から取得手順の記述されたスクリプトを受領し、その手順に従ってプログラム群を取得する。また、クライアント計算機に既に存在するプログラムの場合にはサーバ計算機からの取得は行なわないようにしている。

#### 【0005】

##### 【発明が解決しようとする課題】

本発明の目的は、簡単な操作を行なうだけで、自動的にアプリケーションの実行環境の準備を行い、そのアプリケーションを実行できるコンピュータシステム

を提供することにある。

【0006】

本発明の他の一つの目的は、コンピュータに関する詳しい知識を有していないなくても、所定のアプリケーションを実行できるコンピュータシステムを提供することにある。

【0007】

本発明の他の一つの目的は、アプレットを、アプリケーションの起動時にだけ使用し、アプリケーションのサーバからの取得及び導入を行った後、ローカルで実行できるアプリケーションを起動する事で、サーバ上に存在するアプリケーションを容易かつ高速に起動する事が出来る。

【0008】

【課題を解決するための手段】

インターネット上簡単に配布ができるアプレットをブートストラップとして使用し、そのブラウザが動作しているOS上で直接稼動するアプリケーションを起動する。本発明の一態様においては、アプレットは、単にアプリケーションのコードをダウンロードするだけではなく、ローカル・ファイルシステムにキャッシュされているか否かの判断を行ったり、前提条件となる他のソフトウェアの確認や、そのバージョンのチェックを行う。ローカル・ファイルシステムにキャッシュされている場合には、アプリケーションのダウンロードは行なわない。また、前提条件となる他のソフトウェアが存在しない場合には、そのソフトウェアをダウンロードし、アプリケーションを起動する。

【0009】

本発明の一態様においては、

サーバからロードしたアプリケーションをクライアント・マシン上で起動する方法であって、

(a) 前記クライアント・マシン側でのアプリケーション起動要求に応答して、サーバより実行環境を確認する実行環境確認アプレットをロードする段階と、

(b) 前記実行環境確認アプレットを実行した結果に基づいて、前記クライアント・マシン上で起動するアプリケーションが必要とするコードと前記アプリケ

ーションの起動コマンドをサーバからロードする段階と、

- (c) 前記起動コマンドを前記クライアント・マシンにおいて実行することにより、前記アプリケーションを起動する段階と、  
を含むアプリケーションの起動方法が提供される。

#### 【0010】

なお、本願明細書の特許請求の範囲において、「アプリケーションをロードするサーバ」、「実行環境確認アプレットをロードするサーバ」、「アプリケーションが必要とするコードと前記アプリケーションの起動コマンドをロードするサーバ」は、共通のサーバであっても、各々異なるサーバであってもよい。また、「アプリケーションが必要とするコード」は、そのアプリケーション本体のコードのみならずそのアプリケーションの実行に際し必要となる付随的コードを包括する概念である。

#### 【0011】

本発明の他の一態様においては、

サーバからロードしたアプリケーションをクライアント・マシン上で起動する方法であって、

- (a) 前記クライアント・マシン側でのアプリケーション起動要求に応答して、前記クライアント・マシン上で起動するアプリケーションが必要とするコードと前記アプリケーションの起動コマンドをサーバからロードする段階と、

- (b) 前記起動コマンドを前記クライアント・マシンにおいて実行することにより、前記アプリケーションを起動する段階と、  
を含むアプリケーションの起動方法が提供される。

#### 【0012】

本発明の他の一態様においては、

前記必要とするコードが前記クライアント・マシンに存在するか否かを判断する段階をさらに含み、前記アプリケーションが必要とするコードと前記アプリケーションの起動コマンドは、前記必要とするコードが前記クライアント・マシンに存在していない場合に前記必要とするコードをサーバからロードすることを特徴とするアプリケーションの起動方法が提供される。

【0013】

本発明の他の一態様においては、  
サーバからロードしたアプリケーションをクライアント・マシン上で起動する  
ためのソフトウェア・プロダクトであって、  
(a) 前記クライアント・マシン側でのアプリケーション起動要求に応答して  
、サーバより実行環境を確認するための実行環境確認ソフトウェアと、  
(b) 前記クライアント・マシン上で起動するアプリケーションが必要とする  
コードを取得する必要コード取得ソフトウェアと、  
(c) 前記アプリケーションの起動コマンドを前記クライアント・マシンにお  
いて実行することにより、前記アプリケーションを起動するアプリケーション起  
動ソフトウェアと、  
を含むソフトウェア・プロダクトが提供される。

【0014】

本発明の他の一態様においては、  
サーバからロードしたアプリケーションをクライアント・マシン上で起動する  
ためのソフトウェア・プロダクトであって、  
(a) 前記クライアント・マシン側でのアプリケーション起動要求に応答して  
、前記クライアント・マシン上で起動するアプリケーションが必要とするコード  
と前記アプリケーションの起動コマンドをサーバからロードする必要コード取得  
ソフトウェアと、  
(b) 前記起動コマンドを前記クライアント・マシンにおいて実行することに  
より、前記アプリケーションを起動するアプリケーション起動ソフトウェアと、  
を含むソフトウェア・プロダクトが提供される。

【0015】

本発明の他の一態様においては、  
前記必要コード取得ソフトウェアは、前記必要となるコードが前記クライアン  
ト・マシンに存在するか否かを判断する機能を有していることを特徴とするソフ  
トウェア・プロダクトが提供される。

【0016】

## 【実施例】

## A. ハードウェア構成

図1は、本発明のコンピュータシステム100を構成する情報端末110またはサーバ150を実施するためのハードウェア構成の概観図である。情報端末110またはサーバ150は、中央処理装置(CPU)1とメモリ4とを含んでいる。CPU1とメモリ4は、バス2等を通して、補助記憶装置としてのハードディスク装置13、31とを接続してある。フロッピーディスク装置(またはMO28、CD-ROM26、29等の媒体駆動装置26、28、29、30)20はフロッピーディスクコントローラ(またはIDEコントローラ25、SCSIコントローラ27等の各種コントローラ)19を介してバス2へ接続されている。

## 【0017】

フロッピーディスク装置(またはMO、CD-ROM等の媒体駆動装置26、28、29、30)20には、フロッピーディスク(またはMO、CD-ROM等の媒体)が挿入され、このフロッピーディスク等やハードディスク装置13、ROM14等の記憶媒体には、オペレーティング・システムと協働してCPU等に命令を与え、本発明を実施するためのコンピュータ・プログラムのコードを記録することができ、メモリ4にロードされることによって実行される。このコンピュータ・プログラムのコードは圧縮し、または、複数に分割して、複数の媒体にまたがって記録することもできる。

## 【0018】

情報端末110またはサーバ150は更に、ユーザ・インターフェース・ハードウェアを備えたシステムとことができ、ユーザ・インターフェース・ハードウェアとしては、例えば、画面位置情報を入力をするためのポインティング・デバイス(マウス、ジョイスティック、トラックボール等)7、キー入力をサポートするキーボード6や、イメージデータをユーザに提示するためのディスプレイ11、12がある。また、スピーカ23は、オーディオ・コントローラ21から音声信号を、アンプ22を介して受領し、音声として出力する。

【0019】

この情報端末110またはサーバ150100は、シリアルポート15およびモ뎀または、トークンリング等の通信アダプタ18等を介して、他のコンピュータや通信衛星等と通信を行うことができる。

【0020】

本発明は、通常のパーソナルコンピュータ（PC）やワークステーション、テレビやFAX等の各種家電製品に組み込まれたコンピュータ、車両や航空機等に搭載されるコンピュータ（カーナビゲーション・システム等）又はこれらの組合せによって実施可能である。ただし、これらの構成要素は例示であり、その全ての構成要素が本発明の必須の構成要素となるわけではない。特に、本発明は、アプリケーションの起動方法に関するものであるため、スピーカ23、オーディオ・コントローラ21、アンプ22等の構成要素は本発明の一態様においては必須のものではない。

【0021】

情報端末110のオペレーティング・システムとしては、Windows NT（マイクロソフトの商標）、Windows 9x（マイクロソフトの商標）、Windows 3.x（マイクロソフトの商標）、OS/2（IBMの商標）、Mac OS（Apple社の商標）、Linux（Linus Torvaldsの商標）、AIX（IBMの商標）上のX-WINDOWシステム（MITの商標）などの、標準でGUIマルチウインドウ環境をサポートするものや、PC-DOS（IBMの商標）、MS-DOS（マイクロソフトの商標）などのキャラクタ・ベース環境のもの、さらにはOS/Open（IBMの商標）、VxWorks（Wind River Systems, Inc. の商標）等のリアルタイムOS、Java OS等のネットワークコンピュータに組み込まれたOSでも実現可能であり、特定のオペレーティング・システム環境に限定されるものではない。

【0022】

サーバ150のオペレーティング・システムとしては、Windows NT（マイクロソフトの商標）、Windows 9x（マイクロソフトの商標）、Wi

Windows X（マイクロソフトの商標）、OS/2（IBMの商標）、Mac OS（Apple社の商標）、Linux（Linus Torvaldsの商標）、AIX（IBMの商標）上のX-WINDOWシステム（MITの商標）などの、標準でGUIマルチウインドウ環境をサポートするものが好ましいが、PC-DOS（IBMの商標）、MS-DOS（マイクロソフトの商標）などのキャラクタ・ベース環境のもの、Java OS等のネットワークコンピュータに組み込まれたOSでも実現可能であり、特定のオペレーティング・システム環境に限定されるものではない。

## 【0023】

## B. システム構成

図2は、本発明の好適な実施例におけるコンピュータシステムの処理要素を示す機能ブロック図である。本発明の好適な実施例におけるコンピュータシステム100は、情報端末110と、サーバ150を含んでいる。

## 【0024】

情報端末110には、オペレーティング・システム111と、オペレーティング・システム111上で稼動するクライアント環境120が存在する。クライアント環境120には、ウェブブラウザ121が導入されており、実行時において、ネットワーク140を介してサーバ150からブートストラップ・アプレット123を取得し、Java仮想マシン131上で稼動できるようになっている。

## 【0025】

また、情報端末110は、履歴情報133、起動対象アプリケーション135、起動対象アプリケーション前提ソフトウェア137を有している。起動対象アプリケーション135と起動対象アプリケーション前提ソフトウェア137は、後述する本発明の機能によりサーバ150からダウンロードされる。

## 【0026】

本発明の好適な実施例において、サーバ150からダウンロードされるブートストラップ・アプレット123には、ローカル環境確認手段125、必要コード導入手段127、アプリケーション起動手段129が含まれている。

## 【0027】

ローカル環境確認手段125は、図4に示すローカル環境確認情報を保持し、履歴情報を参照し、導入の必要なコードを決定し、導入が必要なコードがある場合には、必要コード導入手段127にコードの導入を指示する。導入が必要なコードがない場合には、アプリケーション起動手段129にアプリケーションの起動を指示する。

## 【0028】

図3は、本発明の好適な実施例における履歴情報のデータ構造を示す概念図である。本発明の好適な実施例における履歴情報210には、導入済みソフトウェア名211、ソフトウェアID213、導入実行日時215、導入先パス名217、バージョン情報219、前提ソフトウェアID221の情報を含んでいる。ソフトウェアID213、導入実行日時215、導入先パス名217、バージョン情報219、前提ソフトウェアID221の情報は、導入済みソフトウェア名211に対応して登録され、導入済みソフトウェア名221は、複数登録することができる。

## 【0029】

導入済みソフトウェア名211は、「エクセル」、「ワード」（「エクセル」及び「ワード」は、マイクロソフト社の商標）等のアプリケーションやダイナミック・リンク・ライブラリ等のその他のソフトウェア名称であり、ソフトウェアID213は、コンピュータシステム100において、そのソフトウェア固有に付与されて識別子であり、導入実行日時215は、そのソフトウェアを導入した日時の情報である。導入先パス名217は、"c:\windows"等、そのソフトウェアにアクセスするためのパス情報である。バージョン情報219は、そのソフトウェアのバージョン情報である。前提ソフトウェアID221は、そのソフトウェアが動作するために必要なソフトウェアを特定するための情報である。

## 【0030】

図4は、本発明の好適な実施例におけるローカル環境確認情報のデータ構造を示す概念図である。本発明の好適な実施例におけるローカル環境確認情報230

には、導入対象環境情報231、起動アプリケーション名233、アプリケーションID235、導入先パス名237、バージョン情報239、前提ソフトウェアID241、アプリケーション確認手段の情報を含んでいる。ソフトウェアID235、導入先パス名237、バージョン情報239、前提ソフトウェアID241、アプリケーション確認手段243の情報は、導入済みソフトウェア名233に対応して登録され、導入済みソフトウェア名233は、導入対象環境情報231に対し複数登録することができる。また、導入対象環境情報231も複数登録することができる。

## 【0031】

導入対象環境情報231は、「windows98」、「MacOS」（「windows98」は、マイクロソフト社の商標であり、「MacOS」は、アップル社の商標である）等、アプリケーションが動作する環境を特定するための情報である。導入済みアプリケーション名233は、「エクセル」、「ワード」（「エクセル」及び「ワード」は、マイクロソフト社の商標）等のアプリケーションの名称であり、アプリケーションID235は、コンピュータシステム100において、そのソフトウェア固有に付与されて識別子であり、導入先パス名237は、"c:\windows"等、そのアプリケーションにアクセスするためのパス情報である。バージョン情報239は、そのアプリケーションのバージョン情報である。前提ソフトウェアID241は、そのアプリケーションが動作するために必要なソフトウェアを特定するための情報である。アプリケーション確認手段243は、そのアプリケーションが存在することを確認するためのコマンド等の情報である。

## 【0032】

例えば履歴情報133に、

導入済みソフトウェア名 JRE 1. 2. 2

ID 100

導入実行日時 2000/01/01

導入先パス c:\Program Files\JavaSoft\jre\1.2.2

バージョン 1.2.2

前提ソフトウェアID 無し  
が登録されていたとする。

【0033】

このJRE 1.2.2 の存在確認の為、例えば以下のような確認コマンド（スクリプト、疑似コード）を実行する。

「execute "java -version"; (java -version を実行)」。

導入済みの場合、これに対する応答は、「java version "1.2.2"」という文字列が返る。またバージョンを確認するために以下のコマンドも発行する。

「parseVersion "java version ¥" {0}¥"; (応答の文字列を確認)」。

パースに失敗した場合は、確認失敗の結果が返る。そして、

「if (version >= "1.2.2") return ok; (Version が 1.2.2 以上を示していればOK)」、「else return bad;」により、所望のバージョンが存在するか否かを確認することができる。

【0034】

また、履歴情報133に、  
導入済みソフトウェア名 BAP 1.0  
ID 200  
導入実行日時 2000/01/01  
導入先パス c:\BAP  
バージョン 1.0  
前提ソフトウェアID 100  
が登録されていたとする。

【0035】

このBAP 1.0 の存在確認の為、例えば以下のような確認コマンドにより確認することができる。

「cd \$(PATH200); (履歴情報の導入先パスの ID 200 のディレクトリに移動)」、

「execute "java -classpath bap.jar com.ibm.apst.bap.Version" (導入済みの場合、応答は BAP 1.0 という文字列が返る)」。

また、バージョンの確認のため、前述の例と同様に、

```
'parseVersion "BAP {0}";',
'if (version >= "1.0") return ok;',
'else return bad;' を発行する。
```

#### 【0036】

必要コード導入手段127は、ローカル環境確認手段125より取得すべきソフトウェア名、ソフトウェアID、導入先のパス情報、バージョン情報のセットを取得し、サーバ150に対し、そのソフトウェアを取得するためのリクエストを送信する。また取得したソフトウェアを導入先パス情報に基づいて登録し、また登録の確認を行なう。アプリケーション起動手段129は、アプリケーションの起動や起動の確認を行なう。なお、本発明の好適な実施例におけるブートストラップ・アプレットや導入するソフトウェアには認証機関による署名が含まれており、ユーザがブラウザ等においてその認証機関の署名を含むソフトウェアのセキュリティ・チェックを解除する設定にすることによって、セキュリティを維持しつつ、繁雑な確認操作を行なうことなく導入作業が実行されることとなる。

#### 【0037】

図5は、本発明の好適な実施例におけるアプリケーション起動情報のデータ構造を示す概念図である。本発明の好適な実施例におけるアプリケーション起動情報250には、導入対象環境情報251、導入済みアプリケーション名253、アプリケーションID255、起動コマンド257、起動確認手段259の情報を含んでいる。アプリケーションID255、起動コマンド257、起動確認手段259の情報は、導入済みソフトウェア名253に対応して登録され、導入済みソフトウェア名253は、導入対象環境情報251に対し複数登録することができる。また、導入対象環境情報251も複数登録することができる。

#### 【0038】

導入対象環境情報231は、ローカル環境確認情報の場合と同様に「windows98」、「MacOS」等、アプリケーションが動作する環境を特定するための情報である。導入済みアプリケーション名253も、「エクセル」、「ワード」（「エクセル」及び「ワード」は、マイクロソフト社の商標）等のアプリ

ケーションの名称であり、アプリケーションID235は、コンピュータシステム100において、そのソフトウェア固有に付与されて識別子である。起動コマンド257は、そのアプリケーションを起動するためのコマンドの情報である。起動確認手段259は、そのアプリケーションが正常に起動されたか否かを確認するためのコマンド等の情報である。

## 【0039】

なお、前述の例で示したBAP 1.0 の起動に際しては、以下のような起動コマンド（スクリプト、疑似コード等）を実行することができる。

```
「cd $(PATH200); (履歴情報の導入先パスのID200のディレクトリに移動)
「execute "java -classpath bap.jar com.ibm.apst.bap.BAP data¥Start.xml"
」
```

## 【0040】

また、BAP 1.0 の起動確認の為、上記起動コマンド実行後、以下のような起動確認コマンド（スクリプト、疑似コード等）を実行することができる。

```
「findString "BAP Started" till 30 sec; (30秒以内に BAP Started の文字列を標準出力で確認する。確認できない場合、失敗を返す)
「process running; (起動したプロセスは終了していない。終了している場合、失敗を返す)
「return ok;」。
```

## 【0041】

## C. 処理手順

本発明の処理手順を図6に示すフローチャートを用いて説明する。

## 【0042】

まず、クライアント環境120において、ユーザがウェブブラウザ121を起動し、サーバ150のURLを入力することにより（ステップ403）、ブートストラップ・アプレット123がロードされる（ステップ405）。

## 【0043】

図9は、ブートストラップ・アプレット123がロードされた状態の本発明の

好適な実施例におけるウェブ・ブラウザ画面を示す概念図である。図に示すように、ブートストラップ・アプレット123がロードされた状態の本発明の好適な実施例におけるウェブ・ブラウザ画面700には、アプリケーション起動ボタン701が配置される。

#### 【0044】

なお、本発明の好適な実施例においては説明を簡単にするためにローカル環境確認手段125、必要コード導入手段127、アプリケーション起動手段129のコードをアプリケーション起動ボタン701が押されると同時にロードしているが、各コードのロードは、ボタン701のクリックと同時ではなくても良い。例えば、ボタン701が押されると同時にロードされるブートストラップ・アプレット123には、各コードにアクセスするための情報を埋めこんでおき、各コードの実行時または、それに先立って各コード毎、または所定の組み合わせでサーバ150または他のサーバから取得してもよい。

#### 【0045】

ユーザがアプリケーションの起動を要求すべく、アプリケーション起動ボタン701を押すと（ステップ407）、ローカル環境確認手段125が起動する（ステップ409）。図7は、ローカル環境確認手段の主な動作手順を示すフローチャートである。なお、本発明の一態様においては、ユーザによるアプリケーションの起動要求に先立って、アプリケーションの導入先等をユーザに設定されることも可能である。

#### 【0046】

まず、ローカル環境確認手段125は、情報端末110のオペレーティング・システム111の種類を確認し、ローカル環境確認情報230（図4）に登録されたそのオペレーティングシステムに対応する起動アプリケーションを決定し、その情報233～243を取得する（ステップ443）。なお、情報端末110のオペレーティング・システム111は、Javaのシステム・プロパティやHTTPリクエストのエージェント名を参照すること等により識別することができる。

#### 【0047】

次に、ローカル環境確認手段125は、履歴情報210（図3）にアクセスし、起動対象アプリケーション233や前提ソフトウェア241の全てが既に導入されているか否かをアプリケーション名やIDやバージョン等により判断する（ステップ435）。その全てが過去に導入済みであると判断された場合には（ステップ436）、処理を図6のフローに戻す（ステップ447）。過去に導入済みでない（バージョンの不一致等を含む）ソフトウェアが存在すると判断された場合には、ローカル環境確認情報230（図4）よりソフトウェア確認手段243を取得し、それを実行する（ステップ437）。

#### 【0048】

具体的には、前述の確認コマンドを発行し、帰ってきたメッセージでソフトウェアの存在やそのバージョンを調べることや、所定のダイナミック・リンク・ライブラリを探索し、その内容を参照することにより、所定のソフトウェアの存在やそのバージョン情報を取得することができる。これにより、履歴情報210には、登録されていないが、情報端末110に既に存在するソフトウェアを有效地に利用することができる。

#### 【0049】

情報端末110において、使用可能かつ有効なソフトウェアの存在が確認されず、必要なコードの導入が必要であると判断された場合には、必要コード導入手段127を実行し、サーバ150またはその他のサーバから必要なコードを取得する（ステップ441）。

#### 【0050】

図8は、必要コード導入手段の主な動作手順を示すフローチャートである。必要コード導入手段が起動されると、必要となるソフトウェアの導入パッケージをサーバ150または他のサーバから取得する。サーバからの取得が正常に終了しなかった場合には、所定回数のリトライを行い、それでもかかわらず正常終了できなかった場合には、エラー表示を行い、処理を終了する（ステップ455、457、463、465）。導入パッケージの入手処理が正常に終了した場合には、インストーラの起動を行い、ソフトウェアの導入を行なう（ステップ459）。ソフトウェアの導入が正常に終了しなかった場合には、エラー表示を行い、処

理を終了する（ステップ461、463、465）。ソフトウェアの導入が正常に終了した場合には、処理を図7に返す（ステップ467）。

#### 【0051】

具体例を紹介すると、例えば、BAP 1.0 (ID 200) のパッケージが存在しない場合、ブートストラップ・アプレットは、サーバにURLを問い合わせ、`ftp://bootapplet.host/applications/BAP10.exe`を得る。そして、ブートストラップ・アプレットは、適当なローカルディレクトリ、`c:\$BAPPLET\$DOWNLOAD`に `BAP10.exe` をダウンロードする。

#### 【0052】

この例において、入手された、`BAP10.exe`は、自己解凍型の圧縮ファイルであり、`BAP10.exe`を実行する事により、インストーラを含む導入パッケージが現れる。この導入パッケージは、例えば、Install Shield（「Install Shield」は、InstallShield Software Corporationの商標）等で作成された物を利用するともできる。

#### 【0053】

そして解凍されたインストーラを起動する。例えば、上述の「Install Shield」を利用する場合、`setup.exe`をサイレント・モードで応答ファイルを指定して起動する。`(setup -s -f1Response.iss)`。その後、解凍された不要なファイルを消去する。Install Sheildには、解凍から、インストーラの実行、ファイルの消去を、`BAP10.exe`を起動するだけで行うことが出来る導入パッケージを作成するツールも付属している。

#### 【0054】

ステップ439において既に情報端末110に使用可能かつ有効なソフトウェアの存在が確認された場合や、ステップ441において必要なソフトウェアの導入が正常に終了した場合には、その情報を履歴情報210に記録する（ステップ443）。前提となるソフトウェアを含む全てのソフトウェアの導入が確認された場合には処理を図6のフローに戻す（ステップ447）。更にソフトウェアの導入が必要な場合には、ステップ435に処理を戻す。

#### 【0055】

図6のステップ409の処理が終了すると、次に対象アプリケーションが起動される（ステップ413）。具体的には、アプリケーション起動情報250（図5）にアクセスし、対象のアプリケーションに対応する起動コマンド257を実行する。そして、対応する起動確認手段259を実行することにより、対象アプリケーションの起動を確認する。そして、起動が正常に終了したと判断された場合は、本発明の処理を終了し、正常に終了しなかったと判断された場合は、リトライを行なう（ステップ419）。リトライが失敗した場合にはエラー表示を行い、処理を終了する（ステップ421、417）。

#### 【0056】

##### 【発明の効果】

以上説明したように、本発明によれば、簡単な操作を行なうだけで、自動的にアプリケーションの実装環境の準備を行い、そのアプリケーションを実行することができる。

#### 【0057】

本発明の一態様においては、コンピュータに関する詳しい知識を有していなくても、簡単な操作を行なうだけで、ネットワークに存在する所定のアプリケーションを所定の場所に導入し、そのアプリケーションを実行し、利用することができる。

#### 【0058】

本発明の一態様においては、署名付きアプレットを、アプリケーションの起動時にだけ使用し、アプリケーションのサーバからの取得及び導入を行った後、ローカルで実行できるアプリケーションを起動する事で、容易かつ高速に起動する事が出来る。

#### 【0059】

##### 【図面の簡単な説明】

【図1】 本発明の情報端末またはサーバのハードウェア構成の一実施例を示すブロック図である。

【図2】 本発明の好適な実施例におけるコンピュータシステムの処理要素を示す機能ブロック図である。

【図3】 本発明の好適な実施例における履歴情報のデータ構造を示す概念図である。

【図4】 本発明の好適な実施例におけるローカル環境確認情報のデータ構造を示す概念図である。

【図5】 本発明の好適な実施例におけるアプリケーション起動情報のデータ構造を示す概念図である。

【図6】 好適な実施例における本発明の処理手順を示すフローチャートである。

【図7】 本発明の好適な実施例におけるローカル環境確認手段の主な動作手順を示すフローチャートである。

【図8】 本発明の好適な実施例における必要コード導入手段の主な動作手順を示すフローチャートである。

【図9】 ブートストラップ・アプレットがロードされた状態の本発明の好適な実施例におけるウェブ・ブラウザ画面を示す概念図である。

【符号の説明】

100 コンピュータシステム

110 情報端末

111 オペレーティングシステム

120 クライアント

121 ウエブブラウザ

123 ブートストラップ・アプレット

125 ローカル環境確認手段

127 必要コード導入手段

129 アプリケーション起動手段

131 Java仮想マシン

133 履歴情報

135 起動対象アプリケーション

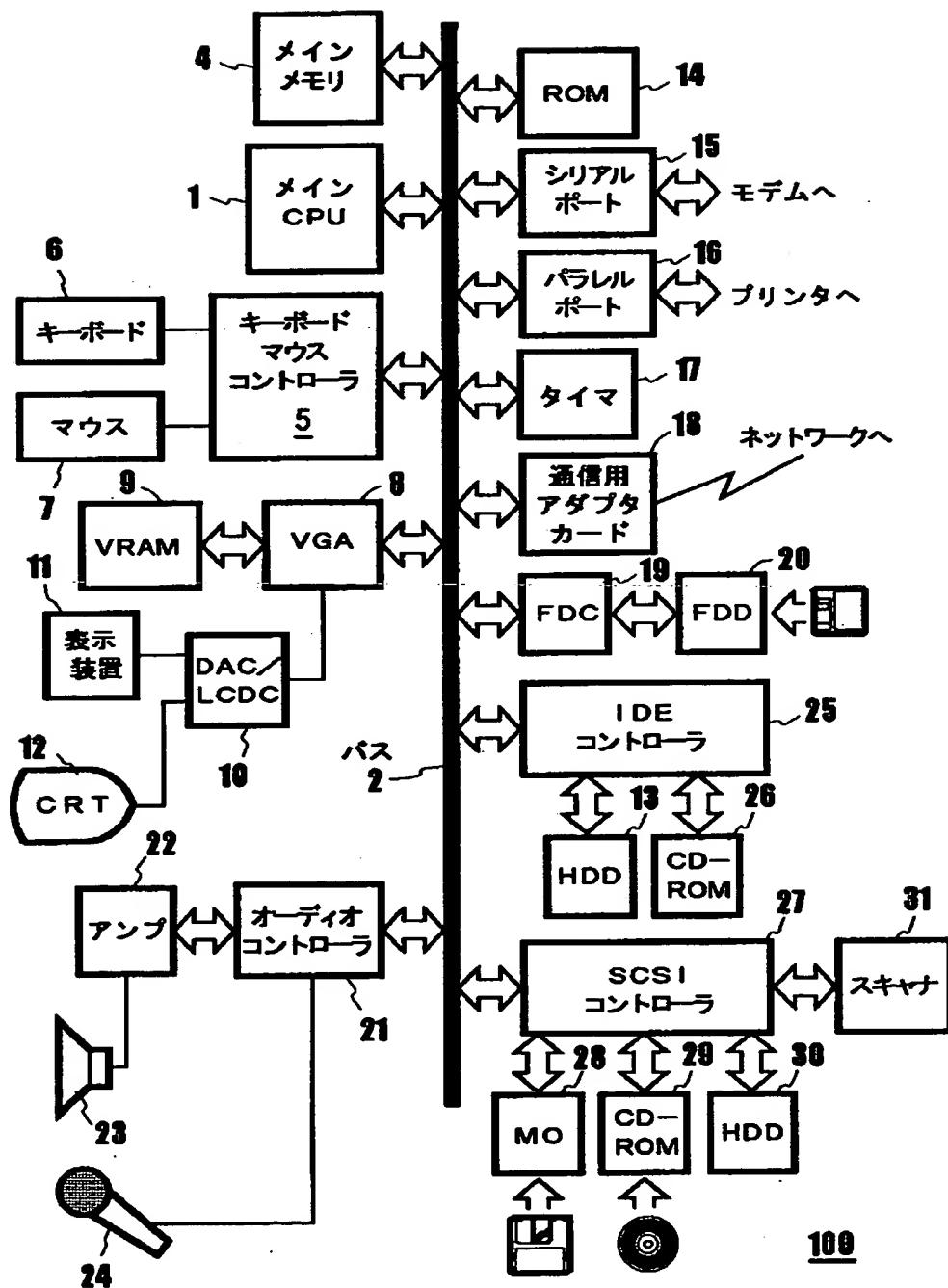
137 起動対象アプリケーション前提ソフトウェア

140 ネットワーク

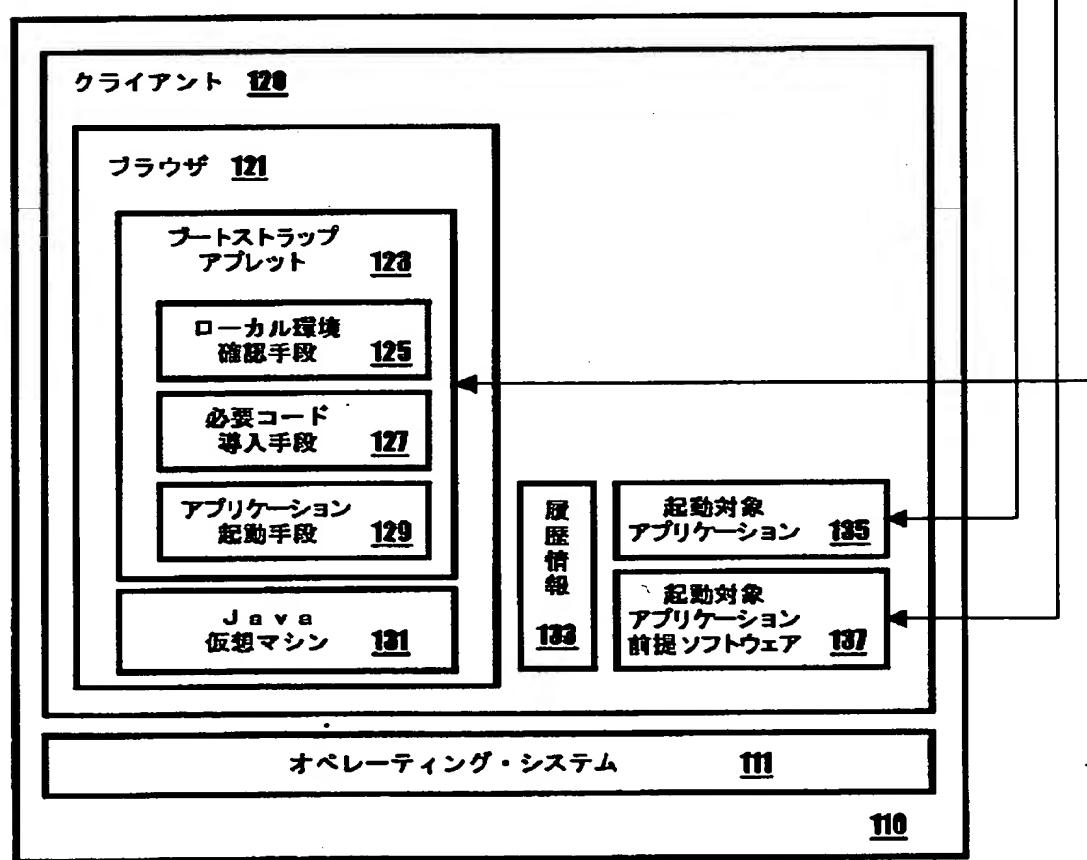
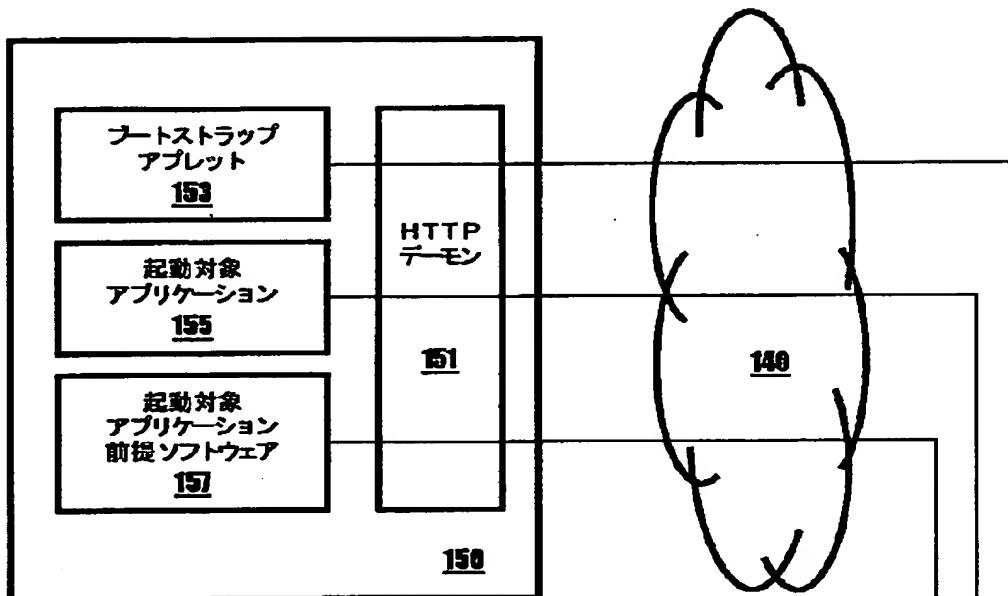
- 150 サーバ・マシン
- 151 HTTPデーモン
- 153 ブートストラップ・アプレット
- 155 起動対象アプリケーション
- 157 起動対象アプリケーション前提ソフトウェア

【書類名】 図面

【図1】



【図2】



【図3】

導入済みソフトウェア	<u>211</u>
ID	<u>213</u>
導入実行日時	<u>215</u>
導入先パス	<u>217</u>
バージョン	<u>219</u>
前提ソフトウェアID	<u>221</u>

210

【図4】

導入対象環境(OSなど)	<u>231</u>
起動アプリケーション	<u>233</u>
ID	<u>235</u>
導入先パス	<u>237</u>
バージョン	<u>239</u>
前提ソフトウェアID	<u>241</u>
アプリケーション確認手段	<u>243</u>

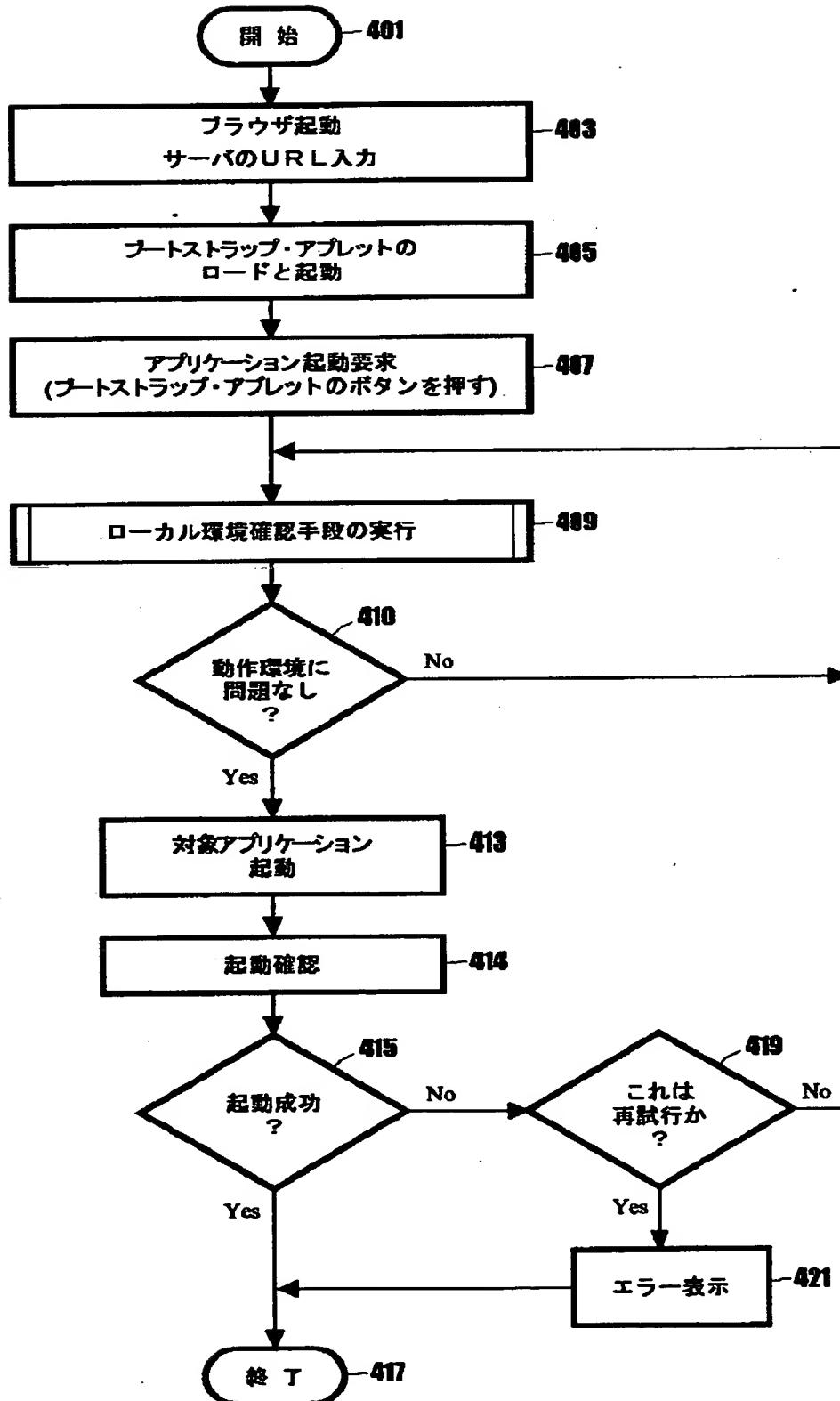
230

【図5】

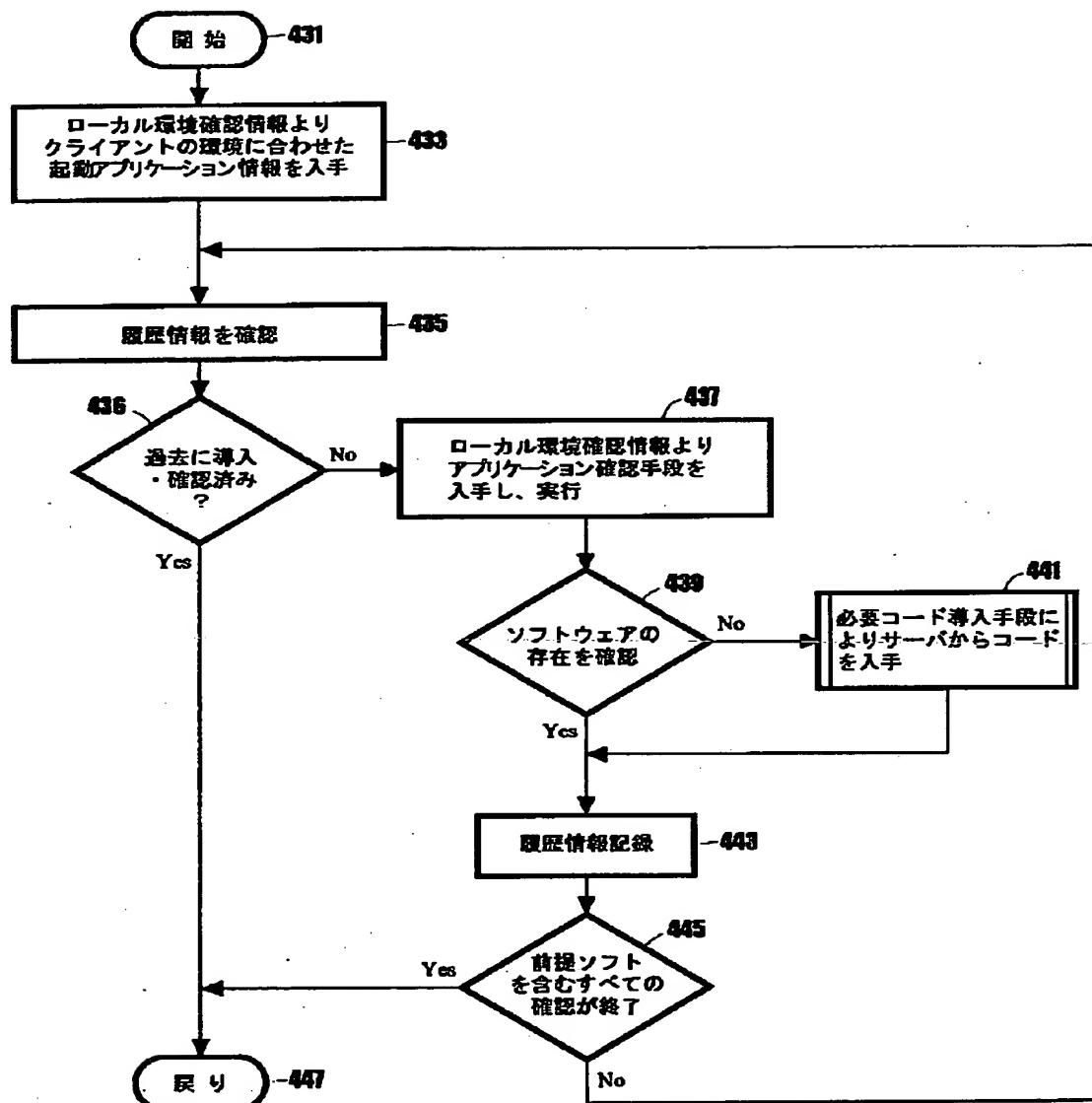
導入対象環境(OSなど)	<u>251</u>
導入済みソフトウェア	<u>253</u>
ID	<u>255</u>
起動コマンド	<u>257</u>

250

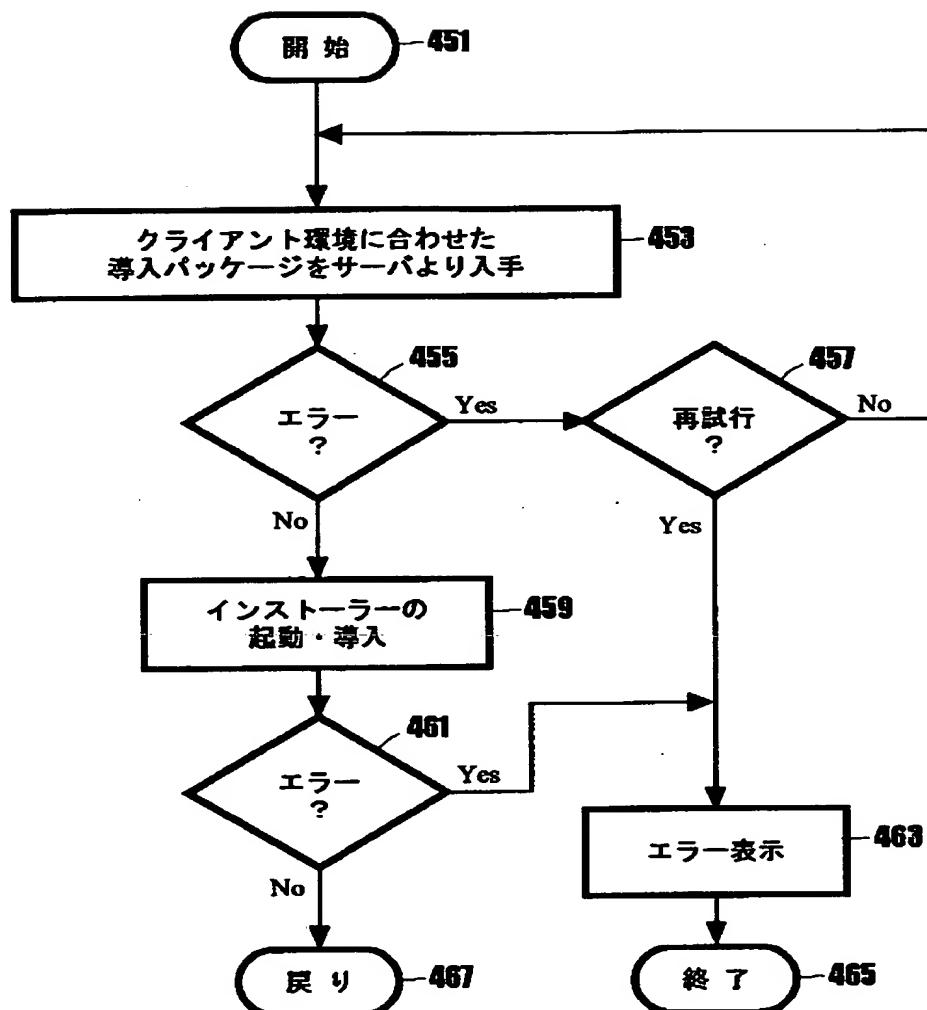
【図6】



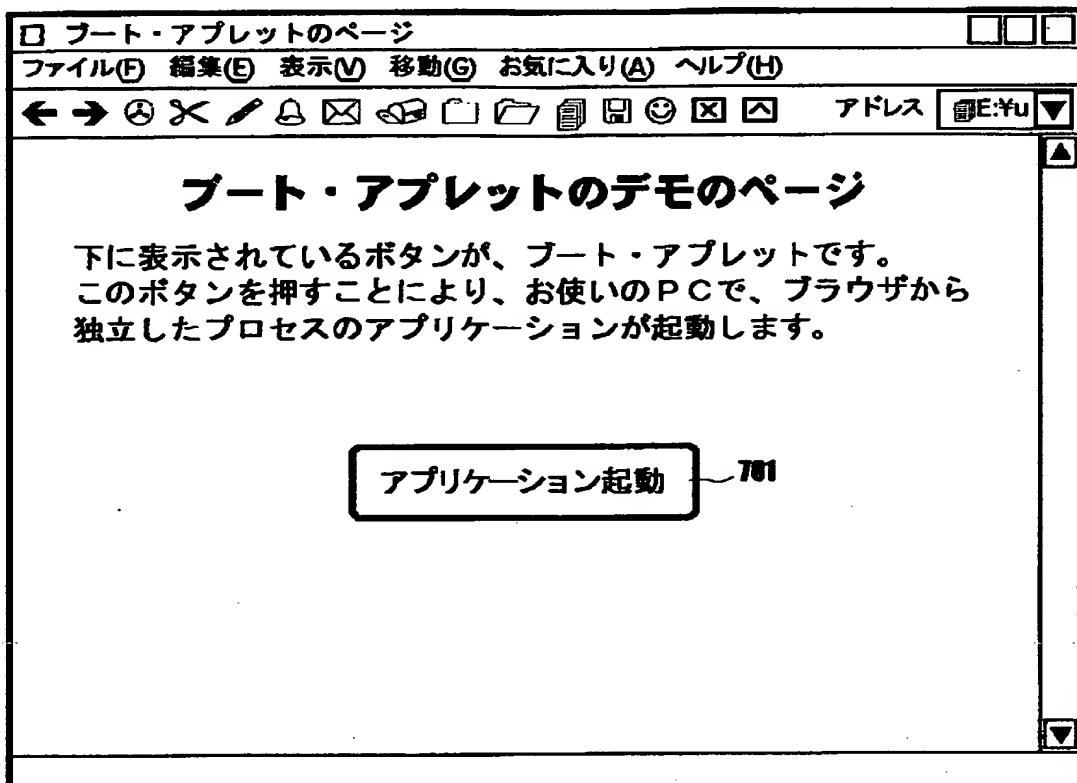
【図7】



【図8】



【図9】



700

【書類名】 要約書

【要約】

【課題】 簡単な操作を行なうだけで、自動的にアプリケーションの実装環境の準備を行い、そのアプリケーションを実行する。

【解決手段】

インターネット上簡単に配布ができるアプレットをブートストラップとして使用し、そのブラウザが動作しているOS上で直接稼動するアプリケーションを起動する。本発明の一態様においては、アプレットは、単にアプリケーションのコードをダウンロードするだけではなく、ローカル・ファイルシステムにキャッシュされているか否かの判断を行ったり、実行環境の確認や、前提条件となる他のソフトウェアの確認、そのバージョンのチェック等を行う。ローカル・ファイルシステムにキャッシュされている場合には、アプリケーションのダウンロードは行なわない。また、前提条件となる他のソフトウェアが存在しない場合には、そのソフトウェアをダウンロードし、アプリケーションを起動する。

【選択図】 図2

認定・付加情報

特許出願の番号 特願2000-001043  
受付番号 50000005521  
書類名 特許願  
担当官 第七担当上席 0096  
作成日 平成12年 1月 7日

<認定情報・付加情報>

【提出日】 平成12年 1月 6日

次頁無

出願人履歴情報

識別番号 [390009531]

1. 変更年月日 1990年10月24日

[変更理由] 新規登録

住 所 アメリカ合衆国10504、ニューヨーク州 アーモンク (番地なし)

氏 名 インターナショナル・ビジネス・マシーンズ・コーポレイション